**THE BIG FOUR OOP BUIDING BLOCKS**

Ở phần này chúng ta sẽ ngâm cứu 4 khái niệm lớn trong khái niệm OOP đó là abstraction, encapsulation, polymorphism and inheritance.

**ABSTRACTION**

Một phần lập trình OOP tốt có liên quan khá nhiều tới abstraction, abstraction không phải là một kỹ thuật lập trình, nó có tác dụng như việc khái niệm báo quát vấn đề mà ta đang giải quyết. Abstraction chia nhỏ vấn đề mà ta đang giải quyết thành những phần nhỏ có thể quản lý được (theo lối suy nghĩ OOP). Design Pattern cũng đồng nghĩa với việc ta dành phần lớn thời gian với abstraction.

**ENCAPSULATION**

Khi chúng ta đóng gói các phương thức và dữ liệu thành một object, chúng ta đã thực hiện việc encapsulation. Nó giúp chúng ta có cái nhìn đơn giản hướng vào một đối tượng đơn giản, bỏ đi những cái nhìn phức tạp liên quan tới nhiều vấn đề khác nhau. Một ví dụ đơn giản, trong nhà bếp có nhiều vật dụng, với cái nhìn tổng quát hết nhà bếp, ta khó lòng nắm bắt được hết, tuy nhiên nếu nhìn dưới phương diện từng thành phần nhỏ, như tủ lạnh, nồi cơm, chén bát … ta có thể đóng gói thành những phần nhỏ hơn và xử lý chúng riêng lẽ một cách dễ dàng hơn.

Khi chúng ta Encapsulation một đối tượng, chúng ta sẽ quyết định interface cho đối tượng đó. Sử dụng interface cho phép ta đơn giản đối tượng. Ví dụ như đối tượng tủ lạnh, nó có rất nhiều vấn đề giải quyết phức tạp khác nhau. Tuy nhiên chúng ta phải quyết định đơn giản hóa nó, quyết định thành phần nào sẽ là public và thành phần nào là thuộc tính. Nói cách khác việc sử dụng interface cho phép ta giấu đi những thành phần nào phức tạp thành những interface đơn giản. Chúng ta nên nhớ phải encapsulation các thành phần thay đổi nhiều, hay các thành phần sử dụng nhiều, lập đi lập lại.

**POLYMORPHISM**

Một khía cạnh quan trọng khác của OOP là tính đa hình. Tính đa hình cho phép ta thực thi các đoạn code mà nó có thể sử dụng cho nhiều đối tượng khác nhau, những đối tượng cho chung bản chất nào đó. Ví dụ như ta muốn viết một hàm sử dụng chung cho các khối hình học, như tam giác, vuông, tròn … Ta có thể áp dụng tính đa hình ở đây (ví dụ 1 hàm vẽ tổng quát cho tất cả các khối hình học). Việc thực thi có thể liên quan nhiều tới các kỹ thuật lập trình OOP nên ta có thể tham khảo các sách hướng dẫn căn bản về OOP (về inheritance)

**INHERITANCE**

Phần cuối trong các khái niệ quan trọng của OOP là Inheritance. Đây là tính thừa kế, nó có tác dụng sử dụng lại được các hàm từ việc thừa kế từ đối tượng cha. Inheritance thường được dùng kết hợp với Polymorphism, nhưng nói về vấn đề này, ta có thể nói nó là composition hơn là inheritance, vì khi đa hình một hàm từ đối tượng cha, ta thực ra đã tạo ra 1 hàm thực thi mới cho lớp con (coi thêm về khái niệm OOP để hiểu rõ). Trong design pattern hay thực tế, ta thường sử dụng composition khá nhiều hơn là khái niệm inheritance. Inheritance được coi là mối quan hệ “is-a”